

《编译原理》研究性学习实验教学训练载体

于二元 徐金安 丁丁 陈钰枫

1 学习目标

编译系统是计算机系统重要的系统软件之一,编译原理课程主要讲授编译系统重要组成部分程序设计语言编译程序的设计原理和实现技术,是计算机科学与技术专业学生的必修专业课程,也是每个优秀的计算机专业人员必修的一门课程,更重要的是,编译原理课程中蕴含着计算机科学中解决问题的思路、抽象问题和解决问题的方法,其内容可让计算机专业学生“享用一辈子”。

正如“编译原理”龙书中所述:编译器设计中有很多通过数学方法抽象出问题本质从而解决现实世界中复杂问题的完美例子,这些例子可以被用来说明如何使用抽象方法来解决:接收一个问题,写出抓住了问题的关键特性的数学抽象表示,并用数学技术来解决它。问题的表达必须根植于对计算机程序特性的深入理解,而解决方法必须使用经验来验证和精化。Compiler designs is full of beautiful examples where complicated real-world problems are solved by abstracting the essence of the problem mathematically. These serve as excellent illustrations of how abstractions can be used to solve problems: take a problem, formulate a mathematical abstraction that captures the key characteristics, and solve it using mathematical techniques. The problem formulation must be grounded in a solid understanding of the characteristics of computer programs, and the solution must be validated and refined empirically.

本课程开展研究性教学的目的是,在教师的引导下以问题回溯与思维启发的方式,使学生在不断的探究过程中掌握编译程序设计和构造的基本原理和实现技术,启迪学生的抽象思维、激发学生的学习兴趣、培养学生的探究精神和专业素养,从而提高学生发现问题、分析问题和解决问题的能力。

2 学习与实践任务

1.1 学习内容

以理论与实践紧密结合的七个研究性学习专题作为教学载体,这七个研究专题是:

专题 1_词法分析程序设计原理与实现技术

专题 2_递归下降语法分析设计原理与实现技术

专题 3_LL(1)语法分析设计原理与实现技术

专题 4_算符优先语法分析设计原理与实现技术

专题 5_基于 SLR(1)分析法的语法制导翻译及中间代码生成程序设计原理与实现技术

专题 6_编译器前端实现（专题 1+专题 5）

专题 7_拓展学习-编译器后端及小型实验语言编译器实现（选做）

1.2 任务要求

1. 以上内容专题 1 至专题 6 为必做；专题 7 为选做；实验占总成绩的 20%。
2. 实验要求设计并完成所做专题，程序应能查看到词法分析的输出二元式序列，语法分析的输出结果、中间代码序列或者汇编语言目标代码，还能给出简单的错误提示。程序设计尽量具有通用性和可扩展性。
3. 编写符合实验语言词法和语法规则的源程序对你的编译程序进行测试，须给出尽可能给出完备的测试用例。
4. 能力培养：深入理解理论对实践的指导作用；基本原理、实现技术和方法的正确运用；编译程序本身的系统性。鼓励同学扩充语言的功能，如增加单词符号的种类，增加赋值语句、条件语句等。

1.3 需提交的文档和程序要求说明

1. 登录本科教学课程平台，按相应文件夹提交分专题提交实验文档等，**请务必注意每个专题的提交截止日期**（具体见课程平台）。

2. 每个专题文件夹命名方式“**学号_姓名_专题 i**”，内含分专题设计和源程序，即【**专题 i 设计（word）**】和【**源程序**】

- 专题 1 至专题 5【**专题 i 设计（word）**】包括：（1）程序功能描述；（2）主要数据结构描述；（3）程序结构描述：设计方法、函数定义及函数之间的调用关系等；（4）程序测试：测试用例、测试结果。
- 专题 6【**专题 6 设计文档（word）**】包括：（1）程序总体结构，与分专题的关系，测试用例和测试结果；（2）学习总结：实验过程中遇到的主要问题、如何克服、对你的程序进行评价、对问题的深入理解和实验的收获。
- 专题 7【**专题 7 设计文档（word）**】包括：（1）编译器后端设计方法和程序功能描述；（2）实验性语言编译器总体结构，与分专题的关系，测试用例和测试结果；（3）学习总结：对编译程序的理解和编译原理课程的建议。

3 学习内容设计

3.1 研究性学习专题 1_词法分析程序设计原理与实现

1. 理论传授

源程序输入与词法分析程序输出的基本方法；正则文法及其状态转换图的基本概念，正则表达式及有限自动机的基本概念；正规文法构造相应的状态转换图的基本方法；正则表达

式构造有限自动机的基本方法及不确定有限自动机确定化的基本方法；词法分析程序的设计与编写。

2. 目标任务

[实验项目] 以下为正则文法所描述的 C 语言子集单词符号的示例，请补充单词符号：
++，--，>>，<<，+=，-=，*=，/=，&&（逻辑与），||（逻辑或），！（逻辑非）等等，给出补充后描述 C 语言子集单词符号的正则文法，设计并实现其词法分析程序。

〈标识符〉→字母 | 〈标识符〉字母 | 〈标识符〉数字

〈无符号整数〉→数字 | 〈无符号整数〉数字

〈单字符分界符〉→+ | - | * | ; | , | (|) | { | }

〈双字符分界符〉→〈大于〉= | 〈小于〉= | 〈小于〉> | 〈感叹号〉= | 〈等于〉= | 〈斜竖〉*

〈小于〉→< 〈等于〉→= 〈大于〉→> 〈斜竖〉→/

〈感叹号〉→!

该语言的保留字：void、int、float、double、if、else、for、do、while 等等（也可补充）。

[设计说明] （1）可将该语言设计成大小写不敏感，也可设计成大小写敏感，用户定义的标识符最长不超过 32 个字符；（2）字母为 a-z A-Z，数字为 0-9；（3）可以对上述文法进行扩充和改造；（4）“/*.....*/”和“//”（一行内）为程序的注释部分。

[设计要求] （1）给出各单词符号的类别编码；（2）词法分析程序应能发现输入串中的错误；（3）词法分析作为单独一遍编写，词法分析结果为二元式序列组成的中间文件；（4）设计两个测试用例（尽可能完备），并给出测试结果。

3. 任务分析 重点解决正则文法到状态转换图的转化问题，词法分析程序的实现。

3.2 研究性学习专题 2_递归下降语法分析设计原理与实现

1. 理论传授

语法分析的设计方法和实现原理；LL（1）文法及其判定；无回溯的递归下降分析的设计与实现；

2. 目标任务

[实验项目] 完成以下描述赋值语句的 LL(1)文法的递归下降分析程序

G[B]: B→V=E

E→TE'

E'→ATE' | ε

T→FT'

$$\begin{aligned}
T' &\rightarrow MFT' \mid \varepsilon \\
F &\rightarrow (E) \mid i \\
A &\rightarrow + \mid - \\
M &\rightarrow * \mid / \\
V &\rightarrow i
\end{aligned}$$

[设计说明] 终结符号 i 为用户定义的简单变量，即标识符的定义。

[设计要求] (1) 输入串应是词法分析的输出二元式序列，即某算术表达式“专题 1”的输出结果，输出为输入串是否为该文法定义的算术表达式的判断结果；(2) 递归下降分析程序应能发现简单的语法错误；(3) 设计两个测试用例（尽可能完备，正确和出错），并给出测试结果；(4) 选做：如有可能，考虑如何用文法描述 C 语言的 if 语句，使整个文法仍然为 LL(1) 文法，并使得你的递归下降程序可以分析赋值语句和 if 语句。

3. 任务分析 重点解决 LL(1) 文法到递归子程序框图的设计。

3.3 研究性学习专题 3_LL(1)语法分析设计原理与实现

1. 理论传授

语法分析的设计方法和实现原理；LL(1) 分析表的构造；LL(1) 分析过程；LL(1) 分析器的构造；

2. 目标任务

[实验项目] 实现 LL(1) 分析中控制程序（表驱动程序）；完成以下描述赋值语句的 LL(1) 文法的 LL(1) 分析过程。

$$\begin{aligned}
G[A]: A &\rightarrow V=E \\
E &\rightarrow TE' \\
E' &\rightarrow ATE' \mid \varepsilon \\
T &\rightarrow FT' \\
T' &\rightarrow MFT' \mid \varepsilon \\
F &\rightarrow (E) \mid i \\
A &\rightarrow + \mid - \\
M &\rightarrow * \mid / \\
V &\rightarrow i
\end{aligned}$$

[设计说明] 终结符号 i 为用户定义的简单变量，即标识符的定义。

[设计要求] (1) 输入串应是词法分析的输出二元式序列，即某算术表达式“专题 1”的输出结果。输出为输入串是否为该文法定义的算术表达式的判断结果；(2) LL(1) 分析过程应能发现输入串出错；(3) 设计两个测试用例（尽可能完备，正确和出错），并给出测试结果；(4) 考虑根据 LL(1) 文法编写程序构造 LL(1) 分析表，并添加到你的 LL(1) 分析程

序中。

4. 任务分析 重点 LL(1) 分析方法和 LL(1) 分析器的实现。

3.4 研究性学习专题 4 算符优先语法分析设计原理与实现

1. 理论传授

语法分析的设计方法和实现原理，算符优先文法、最左素短语、算符优先矩阵、优先函数的基本概念；算符优先文法句型最左素短语的确定；算符优先分析算法的实现；

2. 目标任务

[实验项目] 实现算符优先分析算法，完成以下描述算术表达式的算符优先文法的算符优先分析过程。

$$\begin{aligned} G[E]: E &\rightarrow E+T \mid E-T \mid T \\ T &\rightarrow T * F \mid T / F \mid F \\ F &\rightarrow (E) \mid i \end{aligned}$$

[设计说明] 终结符号 i 为用户定义的简单变量, 即标识符的定义。

[设计要求] (1) 构造该算符优先文法的优先关系矩阵或优先函数；(2) 输入串应是词法分析的输出二元式序列，即某算术表达式“专题 1”的输出结果。输出为输入串是否为该文法定义的算术表达式的判断结果。(3) 算符优先分析过程应能发现输入串出错。(4) 设计两个测试用例（尽可能完备，正确和出错），并给出测试结果；(4) 考虑编写程序根据算符优先文法构造算符优先关系矩阵，并添加到你的算符优先分析程序中。

3. 任务分析 重点解决算符优先矩阵的构造方法和算符优先算法的实现。

3.5 研究性学习专题 5 基于 SLR(1)分析法的语法制导翻译及中间代码生成程序设计原理与实现

1. 理论传授

语法制导的基本概念，目标代码结构分析的基本方法，赋值语句语法制导生成四元式的基本原理和方法，该过程包括语法分析和语义分析过程。

2. 目标任务

[实验项目] 完成以下描述赋值语句 SLR(1)文法语法制导生成中间代码四元式的过程。

$$\begin{aligned} G[A]: A &\rightarrow V=E \\ E &\rightarrow E+T \mid E-T \mid T \\ T &\rightarrow T * F \mid T / F \mid F \end{aligned}$$

$$F \rightarrow (E) \mid i$$
$$V \rightarrow i$$

[设计说明] 终结符号 i 为用户定义的简单变量，即标识符的定义。

[设计要求] (1) 构造文法的 SLR(1) 分析表，设计语法制导翻译过程，给出每一产生式对应的语义动作；(2) 设计中间代码四元式的结构；(3) 输入串应是词法分析的输出二元式序列，即某赋值语句“专题 1”的输出结果，输出为赋值语句的四元式序列中间文件；(4) 设计两个测试用例（尽可能完备），并给出程序执行结果四元式序列。

3. 任务分析 重点解决 SLR(1) 分析表构造，赋值语句文法的改写和语义动作的添加。

3.6 研究性学习专题 6_编译器前端实现

1. 理论传授

编译器前端的组成，理解编译程序的整体概念，语法制导翻译的基本概念和实现方法。

2. 目标任务

[实验项目] 将词法分析程序设计原理与实现（专题 1）和基于 SLR(1) 分析法的语法制导翻译及中间代码生成程序设计原理与实现（专题 5）形成一个程序，即程序的输入为符号串源程序，输出为中间代码四元式序列。

[设计要求] (1) 词法分析的输入为符号串，词法分析的输出为二元式形式的中间文件；(2) 语法制导翻译读取二元式形式的中间文件并生成中间代码四元式形式的中间文件；(3) 设计两个测试用例（尽可能完备），并给出程序执行结果。

3. 任务分析 重点解决编译程序总体结构理解问题；词法分析的输出为语法分析的输入，语法分析和语义分析同时完成，即语法制导翻译。

3.7 研究性扩展学习专题 7-编译器后端及小型实验语言编译器实现（选做）

1. 理论传授

编译器后端的设计，编译器的设计，理解编译程序的整体概念，目标代码生成的基本概念和实现方法。

2. 目标任务

[实验项目] 在专题 6 编译器前端实现的基础上，将所生成的中间代码翻译成某种目标机的汇编代码，实现编译器后端实现的任务，再进一步实现程序的输入是源程序，输出是汇编语言目标代码小型实验语言编译器的任务。生成目标代码的方法可手工编程实现，也可以查找生成工具完成。

[设计要求]（1）编译器后端的输入为中间代码序列，输出为汇编代码序列；（2）小型实验语言编译器的输入是字符串形式的源程序，输出是汇编语言目标程序；（3）设计两个测试用例（尽可能完备），并给出程序执行结果。

3. 任务分析 可通过资料查找的方式，研究和了解编译器后端的手工编程实现方式及借助生成工具的实现方式，拓展学习目前各大公司程序设计语言编译程序的实现方式，深入理解编译原理和编译程序总体结构，提高驾驭程序设计语言的能力，提高复杂问题求解能力，提高工程实践能力。

4 学习考核形式

实验考核采取当面演示和质疑、设计文档检查相结合的方式。实验检查时间见课堂或课程平台通知。